Final Report: June 1993

Asynchronous Design for Parallel Processing Architectures

Contract No.: N00014-89-J-3036
Principal Investigator: Teresa H.-Y. Meng
CIS 132, Stanford University, Stanford, CA. 94305
Phone Number: (415) 725-3636
E-Mail Address: meng@tilden.stanford.edu

e integration of VLSI systems becomes larger and denser, implementing a high-performance parallel architecture using a global clock is becoming more inefficient and difficult to design, primarily due to concerns related to clock skew and data synchronization. To solve this problem, a large system may be broken up into asynchronously communicating processors. Each processor can be either asynchronous or synchronous and therefore execute at a rate best suited for individual tasks. Using such asynchronous communication provides a modular design environment in which processors can be individually optimized to yield prominent performance improvement.

However the advantages of asynchronous design do not come without a cost, due to two-way communication and design restrictions necessary to avoid hazards and races. In addition, adopting an asynchronous design style requires new methods for synthesis and verification. This research work addressed these methods to simplify the design methodology for asynchronous processors and improve their performance.

To provide similar easy-to-use synthesis tools for the design of asynchronous systems, as are available and widely used in the design of synchronous systems, we have developed CAD tools that allow automated design of synchronous, asynchronous, and mixed synchronous/ asynchronous circuits. Our work can be summarized in three major topics: automated gate-level synthesis of asynchronous circuits, a uniform approach to both synchronous and asynchronous circuit synthesis, and efficient verification.

Automated Gate-Level Synthesis of Asynchronous Circuits

Our first research thrust has been the synthesis of efficient hazard-free circuits using standard cell libraries and gate-arrays. We have automated our synthesis procedure in a CAD tool *SYN* for the synthesis of speed-independent circuits, a class of asynchronous circuits [1]. Speed-independent circuits are very robust to gate delay variations and are self-checking for output stuck-at-faults [2]. They are guaranteed to be hazard-free and do not utilize delay-elements to eliminate hazards. *SYN* begins with a state graph specification of the circuit and produces a netlist of basic gates.

SYN's state graph specification provides flexibility for describing circuit behavior because state graphs are a fundamental structure that can easily be derived from any high-level specification language. For example, we have automated the generation of state graphs from specifications using burst-mode finite state machines and signal transition graphs.

We have also incorporated into *SYN* the use observability-don't-cares states to guide logic transformations that optimize for area and speed while maintaining speed-independence [4]. These optimizing logic transformations include gate symmetrization, gate sharing, gate merging, and redundant input removal. On average, they contribute over a 25% delay improvement and a 40% area reduction over the initially synthesized netlist.

We have used *SYN* to generate netlists for a large benchmark of over 30 specifications from industry and academia and found that our optimized circuits are on average 25% faster with an area penalty of only 15% when compared to comparable speed-dependent circuits in which delay elements were added to ensure gate-level hazard-freedom. These surprising results can be attributed primarily to the cost of adding delay elements to the critical path to ensure hazard-freedom,

as opposed to using logic synthesis to guarantee speed-independence as adopted in our design style. Such delay elements are particularly problematic because they may have widely varying real-time performance due to VLSI processing variations. This means designers must resort to designing for the worst case parameters, translating to an increased average delay for such circuits.

SYN, however, at times is forced to use basic gates with large-fanin to ensure speed-independence. These large-fanin gates may not exist in the cell library used and hence need special attention for complete ASIC synthesis. To solve this problem, we have developed a heuristic algorithm which breaks up these large-fanin gates into smaller-fanin gates. This is a difficult problem because, if not done correctly, breaking up gates in general introduces hazards into the circuit [4]. We have found that speed-independence can be maintained by introducing feedback connections, called *acknowledgement wire forks*, to the newly added internal signals of the smaller-fanin gates. The choices of connections to maintain speed-independence influence the circuit size and its speed. Our approach is to add connections that minimize a cost function based on the designer's requirements on area, delay and reliability.

In summary, *SYN* provides logic designers with an automated way to build efficient and robust hazard-free netlists that for most specifications are immediately suitable for standard-cell or gate-array implementations. This provides designers with an easy-to-obtain, robust, self-checking, efficient design choice from a number of high-level languages. Using *SYN*, the designer is freed from doing a detailed hazard analysis and need not rely on delay elements to ensure correct circuit behavior.

A Uniform Approach to the Synthesis of Synchronous and Asynchronous Circuits

Our second research thrust has been the development of a synthesis procedure that can be applied in a uniform way to the design of both synchronous and asynchronous circuits. On the one hand, synchronous design tools do not utilize environmental constraints which are necessary for the design of asynchronous circuits. Also, synchronous tools make no effort to remove hazards and races which cause erratic behavior in an asynchronous circuit. On the other hand, asynchronous design tools do not incorporate timing analysis and cannot easily handle timing constraints which are inherent in synchronous circuits.

We introduce a new synthesis procedure for the design of timed circuits [6]. A timed circuit is specified in the form of a constraint graph which includes both causal relationships, such as environmental constraints, and timing constraints. We propose that the set of circuits specified with timing constraints (timed circuits) is actually a superset of both synchronous and asynchronous circuits. From the timed circuit specification, we systematically derive a hazard-free complex-gate circuit implementation. In addition to developing synthesis algorithms, we also developed a timing analysis algorithm which is used to guide the synthesis procedure. The entire synthesis procedure has been automated in a CAD tool *ATACS*.

In addition to providing a uniform approach, we have shown that our synthesis procedure actually produces better circuits in terms of area and performance than those produced with other tools. In [6], we applied our synthesis procedure to the design of asynchronous circuits. In several practical examples, we found that significant reductions in circuit complexity can be achieved using conservative timing constraints. In particular, in a memory management unit, we were able to reduce the circuit area and delay by over 50 percent over the speed-independent implementation. Our improvement comes from the fact that when timing constraints are considered the system is found to contain fewer states, and thus, less circuitry is needed to avoid hazards. As in the synthesis of speed-independent circuits, we found that delay elements are not necessary in the synthesis procedure, as the timing constraints are introduced as *opportunities* for circuit optimization and hazard-freedom is ensure by logic synthesis, rather than by delaying signal transmission.

Since we incorporate a timing model into the synthesis, we can handle mixed synchronous/asynchronous circuits such as a DRAM controller which interfaces an asynchronous RAM with a synchronous environment. For the DRAM controller example, our timed circuit implementation is at least twice as fast as an equivalent synchronous implementation and 25 percent smaller and faster than a previously published asynchronous implementation. This result can be attributed to our use of explicit timing constraints and complex state-holding gates.

Recently, we have applied our synthesis procedure to the design of synchronous circuits [7]. By treating the clock signal as just another input, we can specify and synthesize synchronous circuits using the same synthesis procedure. We found that by utilizing environmental constraints, our procedure results in circuits that are 2 to 3 times smaller and faster when compared with circuits synthesized using academic and industrial synchronous design tools such as Berkeley's *SIS* and *SYNOPSYS* [8]. These surprising results can be attributed to two features of our synthesis procedure. First, the logic is reduced by utilizing sequential state information to guide synthesis, a feature not incorporated into synchronous synthesis procedures. Second, each signal is implemented using a single complex gate, reducing the delay along the critical path. Utilizing our state information, we make these complex gates state-holding, and thus, they can also be very compact.

To summarize, our synthesis procedure provides designers with a uniform synthesis procedure for the design of both synchronous and asynchronous circuits. This allows designers to postpone decisions on system timing until later in the design cycle. It also facilitates the design of mixed synchronous/asynchronous systems. Finally, by taking advantage of timing constraints in asynchronous design and environmental constraints in synchronous design, our procedure results in circuit implementations which are 2 to 3 times smaller and faster than implementations derived using other synthesis procedures.

Automated Verification

Automated verification of asynchronous circuits has been proven to be a necessary component of a design system. For example, Dill's verifier for speed-independent circuits, *AVER*, and Burch's verifier for timed circuits have found hazards in published circuits and help identify bugs in various asynchronous CAD synthesis software. Existing verifiers, however, cannot handle big circuits and can easily take over 10 minutes to verify a circuit of about 100 gates.

We used the intuition gained from our synthesis system to generate an efficient verifier for the verification of speed-independent circuits which has exponentially smaller run-time complexity than *AVER* [4], resulting in orders of magnitudes saving in computation time on verifying large circuits.

We achieve this complexity reduction by inferring the behavior of internal signals from the circuit netlist and the behavior of primary outputs. We avoid explicitly modeling all behaviors of the internal signals, and thereby circumventing the state explosion problem faced by *AVER*.

Our verifier tests that the circuit is hazard-free and that its behavior satisfies its specification. However, our verifier is conservative and may produce false negatives, but none were found in our large benchmark set of specifications and netlists from academia and industry. Therefore, even though conservative, our verification is tight enough to be useful in practice. This verifier has been an indispensable tool in supporting the design of hazard-free asynchronous circuits, to aid the synthesis process and verify the synthesized circuits to be correct.

Using our verifier, logic designers need not do extensive simulations to verify the correctness of an automatically generated or hand-designed netlist. Simulation is based on single delay values, but verification handles all possible gate delays, thereby freeing the designer from the worry that the circuit will fail under slightly different operating conditions than assumed for checking the circuit's correctness.

Summary

In summary, in the past three years, the results from this project have advanced the the design and synthesis of asynchronous circuits from theoretical pursuits to the domain of practicality and high performance. We established the first standard-cell based (or ASIC) design procedure for implementing asynchronous circuits, derived maybe the only well-established testability property of speed-independent circuits, provided the first uniform approach to efficient synthesis of both synchronous and asynchronous circuits, and developed the most efficient verification tool available today.

This project has supported two Ph.D. students for a period of three years and sponsored more than ten publications in technical journals and international conferences.

References:

[1]. Peter A. Beerel and Teresa H.-Y. Meng, "Automated Gate-Level Synthesis of Speed-Independent Control Circuits", *1992 IEEE International Conference on Computer-Aided Design Digest of Technical Papers*, March 1992.

[2]. Peter A. Beerel and Teresa H.-Y. Meng, "Semi-Modularity and Testability of Speed-Independent Circuits", *Integration, the VLSI Journal*, Sept. 1992.

[3]. Peter A. Beerel and Teresa H.-Y. Meng, "Efficient Verification of Determinate Speed-Independent Circuits", submitted to *IEEE ICCAD '93*, April 1993.

[4]. Peter A. Beerel and Teresa H.-Y. Meng, "Logic Transformations and Observability Don't Cares in Speed-Independent Circuits" submitted to *1993 ACM International Workshop on Timing Issues in the Specification and Synthesis of Digital Systems*, April 1993.

[5]. Chris J. Myers and Teresa H.-Y. Meng, "Synthesis of Timed Asynchronous Circuits", *Proc of IEEE International Conference on Computer Design*, October 1992.

[6]. Chris J. Myers and Teresa H.-Y. Meng, "Synthesis of Timed Asynchronous Circuits", to appear *IEEE Transactions on VLSI Systems*, June 1993.

[7]. Chris J. Myers and Teresa H.-Y. Meng, "A Uniform Approach to the Synthesis of Synchronous and Asynchronous Circuits", submitted to *IEEE ICCAD '93*, April 1993.

[8]. SYNOPSYS. "Version 2.2 Design Complier Reference Manual". Synopsys, Inc., 1992.

REPORT OF INVENTIONS AND SUBCONTRACTS

(Pursuant to "Patent Rights" Contract Clause) (See Instructions on Reverse Side.)

Form Approved
OMB No. 0704-0297
Expires Jun 30, 1992

1a. NAME OF CONTRACTOR / SUBCONTRACTOR	c. CONTRACT NUMBER	2a. NAME OF GOVERNMENT PRIME CONTRACTOR	c. CONTRACT NUMBER	3. TYPE OF REPORT (X one)
Stanford University	N00014-89-J-3036	N/A	N/A	a. INTERIM [X] b. FINAL
b. ADDRESS (Include ZIP Code) Teresa H. Meng, Stanford U. CIS 132,M/C4070, Stanford, CA 94305-4070	d. AWARD DATE (YYMMDD) 7/1/89 - 6/30/93	b. ADDRESS (Include ZIP Code)	d. AWARD DATE (YYMMDD)	4. REPORTING PERIOD (YYMMDD) a. FROM b. TO

SECTION I - SUBJECT INVENTIONS

5. "SUBJECT INVENTIONS" REQUIRED TO BE REPORTED BY CONTRACTOR / SUBCONTRACTOR (If "None," so state)

a. NAME(S) OF INVENTOR(S) (Last, First, MI)	b. TITLE OF INVENTION(S)	c. DISCLOSURE NO., PATENT APPLICATION SERIAL NO. OR PATENT NO.	d. ELECTION TO FILE PATENT APPLICATIONS (1) United States (a) Yes	(b) No	(2) Foreign (a) Yes	(b) No	e. CONFIRMATORY INSTRUMENT OR ASSIGNMENT FORWARDED TO CONTRACTING OFFICER (1) Yes	(2) No
	None							

f. EMPLOYER OF INVENTOR(S) NOT EMPLOYED BY CONTRACTOR / SUBCONTRACTOR		g. ELECTED FOREIGN COUNTRIES IN WHICH A PATENT APPLICATION WILL BE FILED	
(1) (a) Name of Inventor (Last, First, MI)	(2) (a) Name of Inventor (Last, First, MI)	(1) Title of Invention	(2) Foreign Countries of Patent Application
(b) Name of Employer	(b) Name of Employer		
(c) Address of Employer (Include ZIP Code)	(c) Address of Employer (Include ZIP Code)		

SECTION II - SUBCONTRACTS (Containing a "Patent Rights" clause)

6. SUBCONTRACTS AWARDED BY CONTRACTOR / SUBCONTRACTOR (If "None," so state)

a. NAME OF SUBCONTRACTOR(S)	b. ADDRESS (Include ZIP Code)	c. SUBCONTRACT NO.(S)	d. DFAR "PATENT RIGHTS" (1) Clause Number	(2) Date (YYMM)	e. DESCRIPTION OF WORK TO BE PERFORMED UNDER SUBCONTRACT(S)	f. SUBCONTRACT DATES (YYMMDD) (1) Award	(2) Estimated Completion
	None						

SECTION III - CERTIFICATION

7. CERTIFICATION OF REPORT BY CONTRACTOR / SUBCONTRACTOR (Not required if [ ] Small Business or [ ] Non-Profit organization.) (X appropriate box)

a. NAME OF AUTHORIZED CONTRACTOR / SUBCONTRACTOR OFFICIAL (Last, First, MI)	c. I certify that the reporting party has procedures for prompt identification and timely disclosure of "Subject Inventions," that such procedures have been followed and that all "Subject Inventions" have been reported.	
Ruth Kaempf		
b. TITLE Senior Contract Officer	d. SIGNATURE Kaempf	e. DATE SIGNED 6/25/93

DD Form 882, OCT 89 — *Previous editions are obsolete.* — 971/284